

Capitolo 4

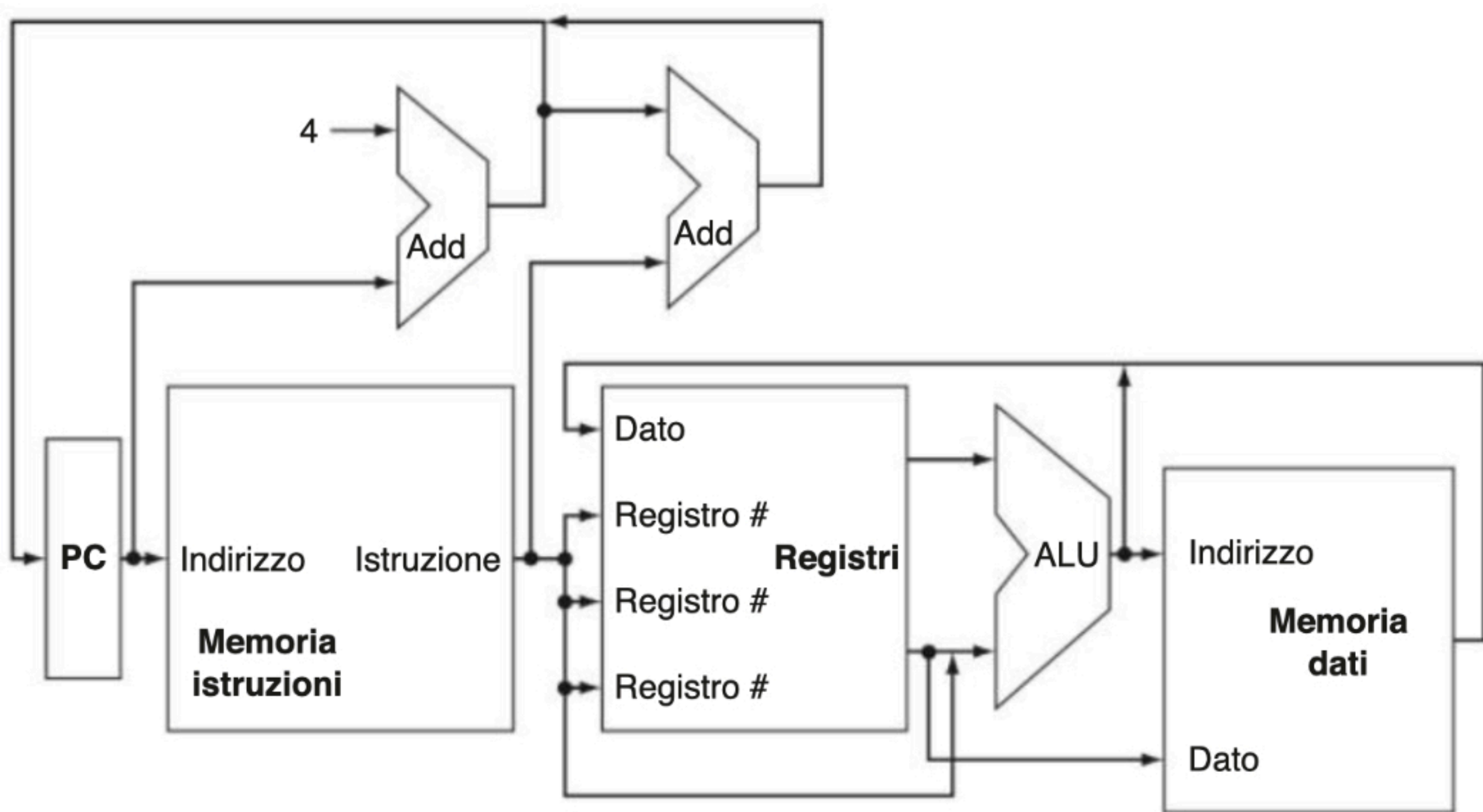
Introduzione

- Fattori delle prestazioni della CPU
 - Instruction count
 - Determinato dall'ISA e dal compilatore
 - CPI e periodo di clock
 - Determinato dall'hardware della CPU
- Studieremo due implementazioni del MIPS
 - Una versione semplificata
 - Una versione più realistica in pipeline
- Semplice sottoinsieme di istruzioni
 - Accesso alla memoria: `lw`, `sw`
 - Operazioni aritmetico/logiche: `add`, `sub`, `and`, `or`, `slt`
 - Controllo del flusso: `beq`, `j`

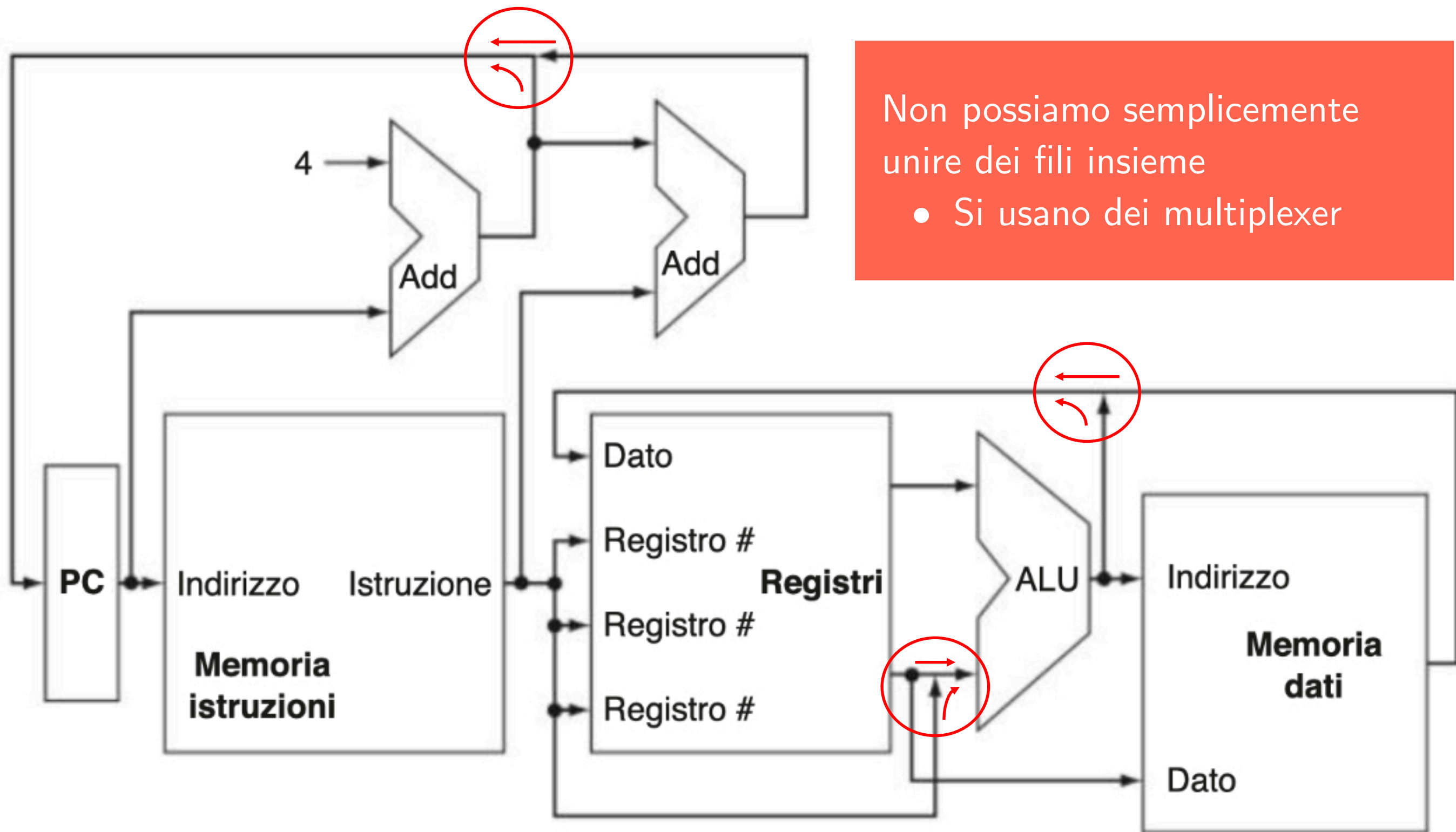
Esecuzione di un'istruzione

- PC \rightarrow memoria istruzioni, preleva (fetch) l'istruzione
- Numeri di registro \rightarrow accesso ai registri, lettura dei registri
- A seconda della classe di istruzioni
 - Usare la ALU per calcolare
 - Un risultato aritmetico
 - L'indirizzo di memoria di una load/store
 - Indirizzo di destinazione di un branch
 - Accedere alla memoria dati per load/store
 - PC \leftarrow indirizzo di destinazione o PC + 4

Panoramica della CPU



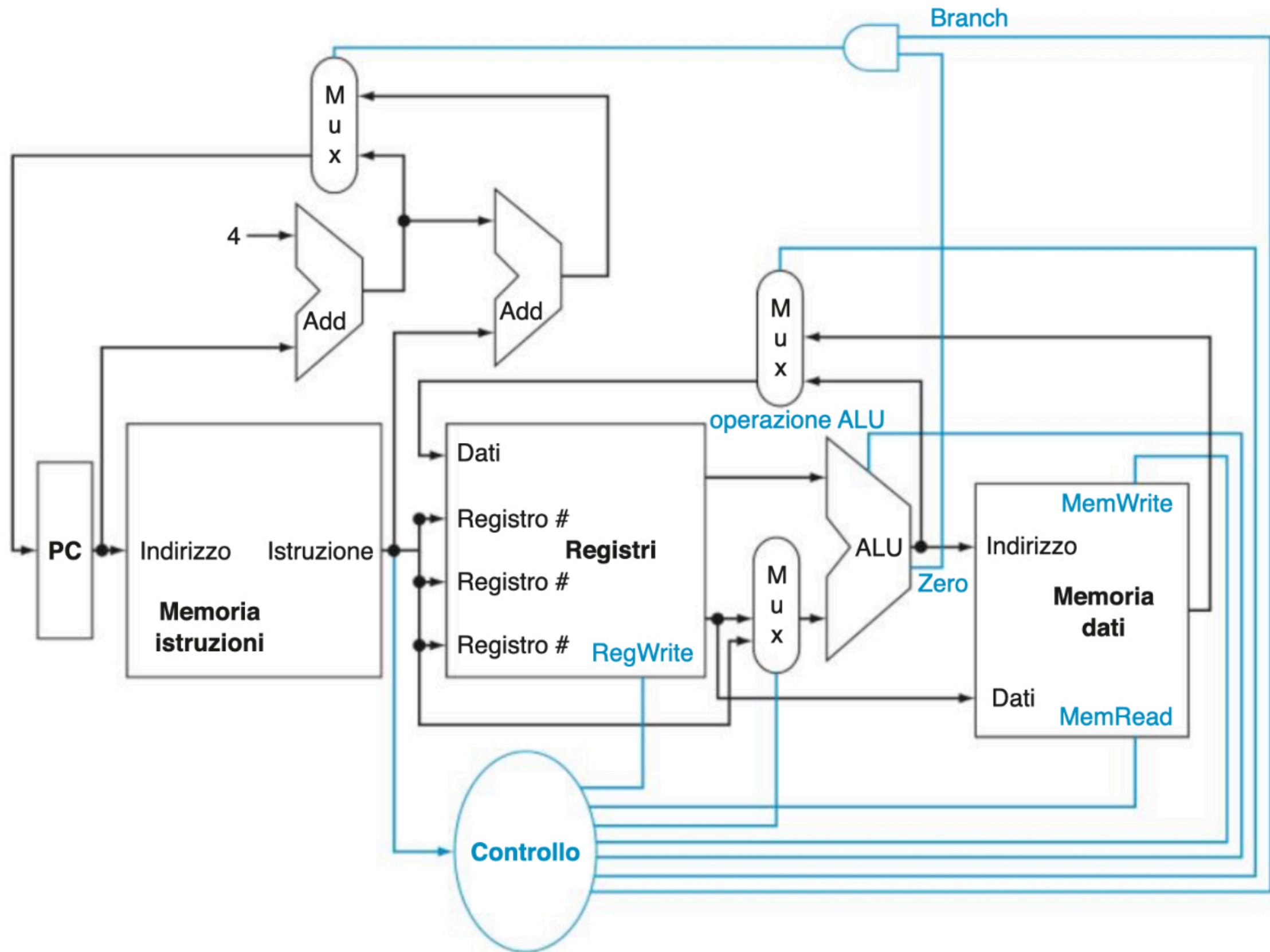
Multiplexer



Non possiamo semplicemente unire dei fili insieme

- Si usano dei multiplexer

Controllo

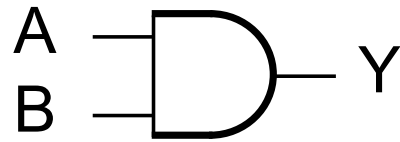


Basi di progettazione logica

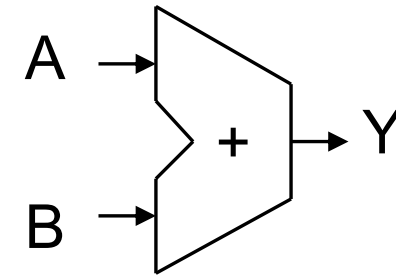
- L'informazione è codificata in binario
 - Tensione bassa = 0, Tensione alta = 1
 - Un filo per bit
 - Dati su più bit codificati su bus multi-filo
- Elemento combinatorio
 - Opera sui dati
 - L'uscita è una funzione dell'ingresso
- Elemento sequenziale (stato)
 - Memorizza l'informazione

Elementi combinatori

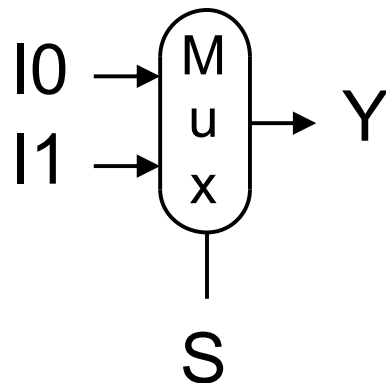
Porta AND
 $Y = A \& B$



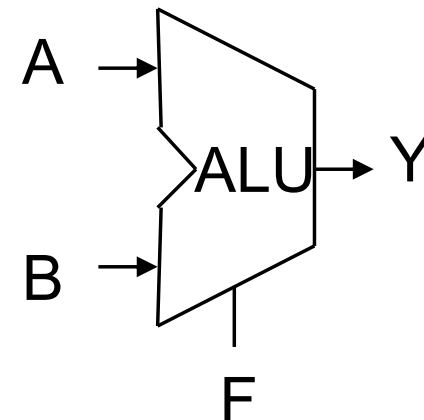
Sommatore
 $Y = A + B$



Multiplexer
 $Y = S ? I1 : I0$

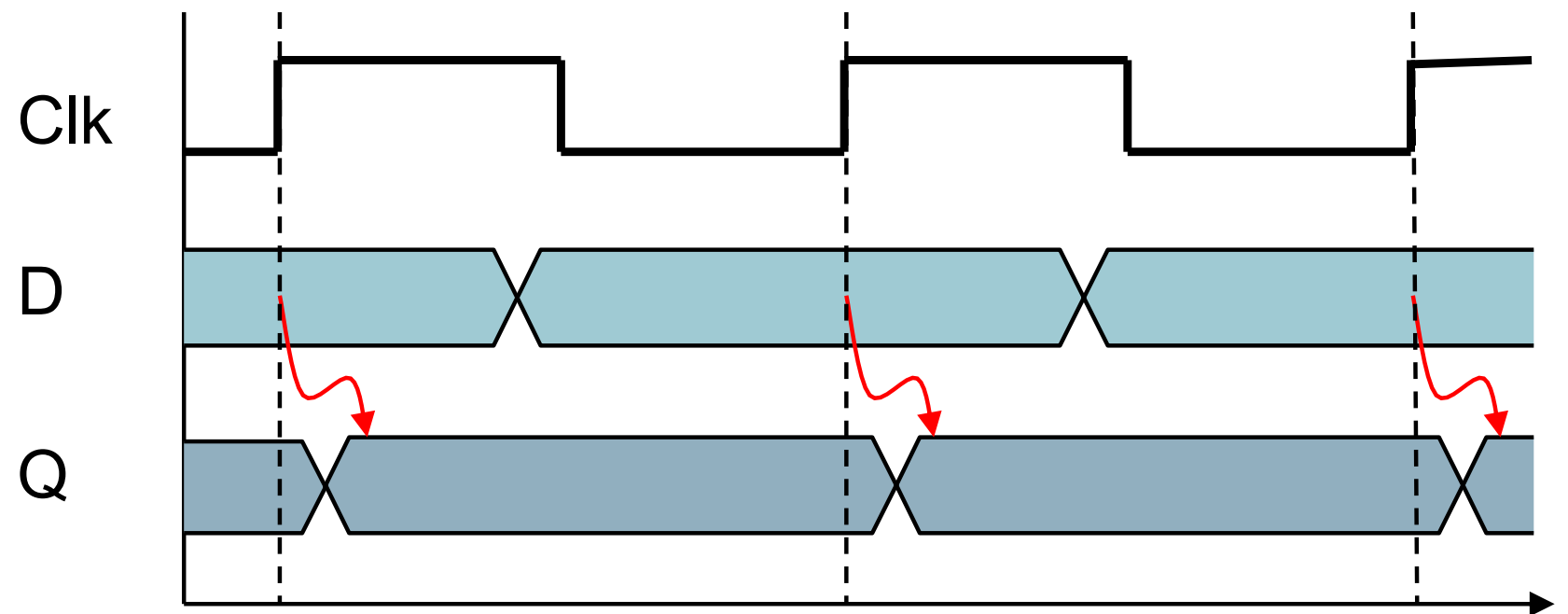
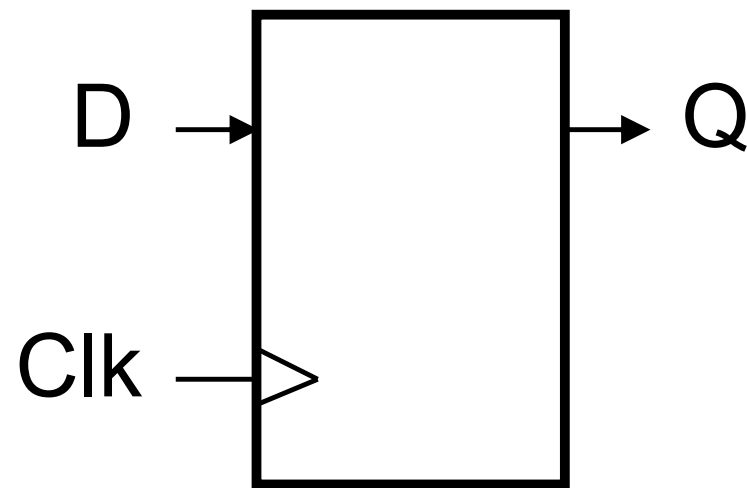


Unità Logico/Aritmetica
 $Y = F(A,B)$



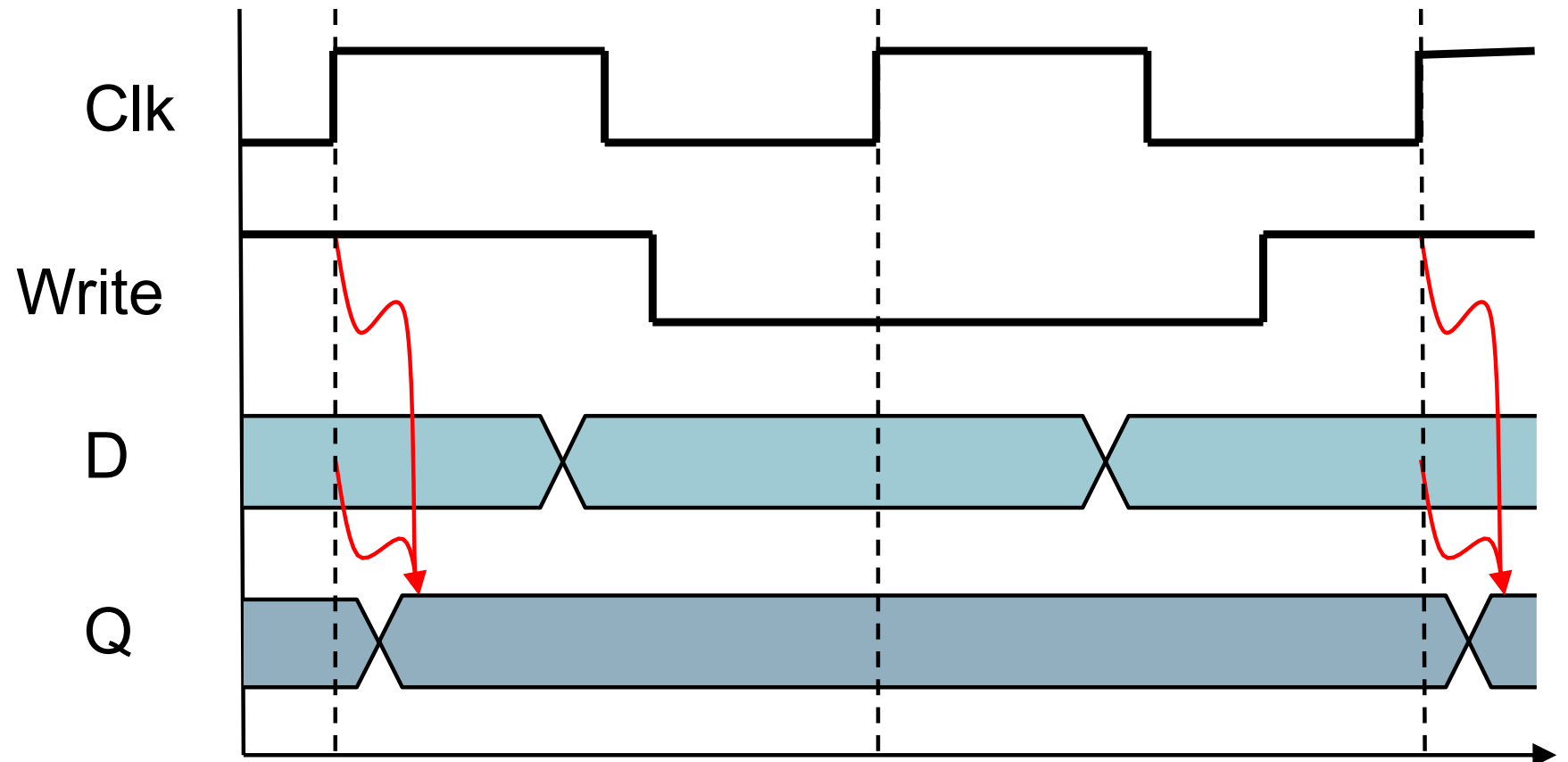
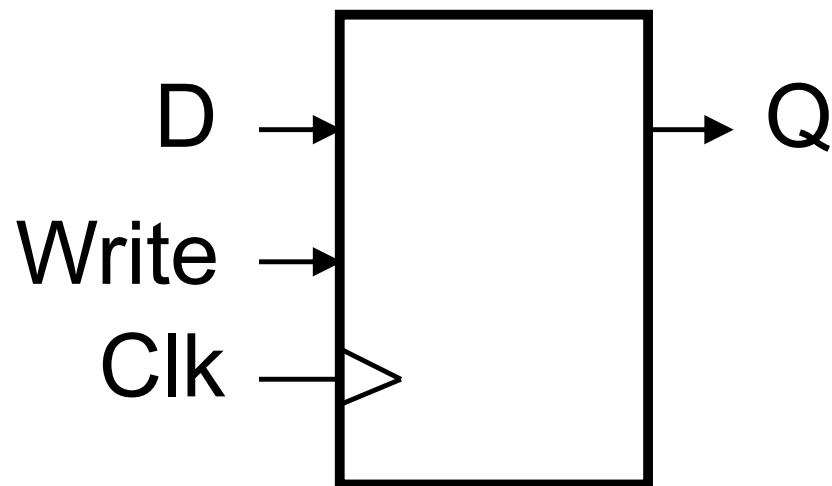
Elementi sequenziali

- Registro: memorizza dati in un circuito
 - Usa il segnale di clock per determinare quando aggiornare il valore memorizzato
 - *Edge-triggered*: aggiorna quando Clk cambia da 0 a 1



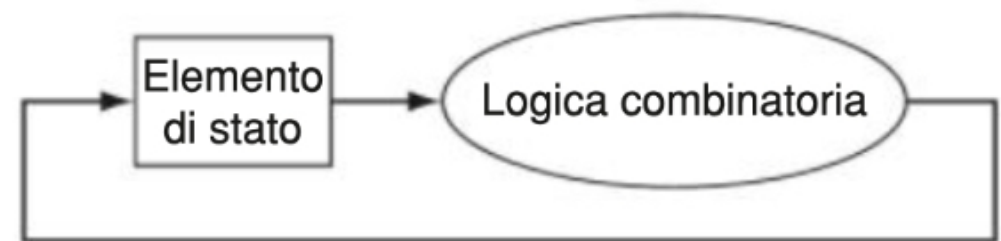
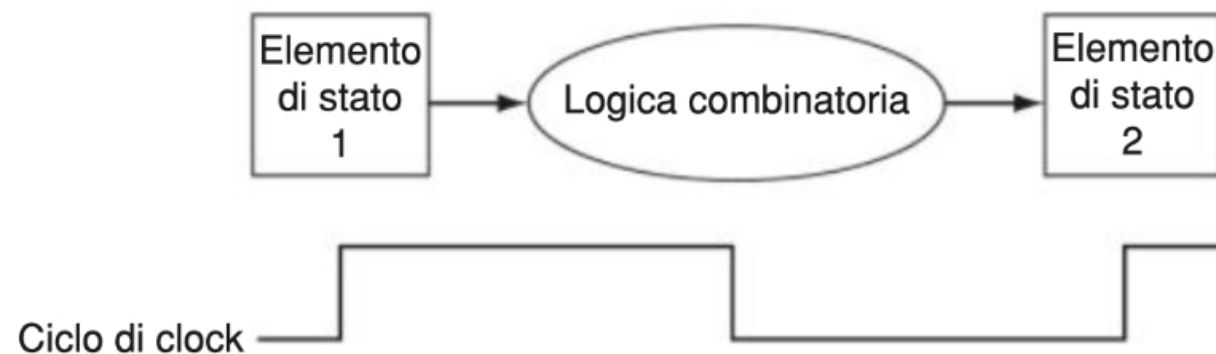
Elementi sequenziali

- Registro con controllo della scrittura
 - Aggiorna unicamente sul fronte del clock quando il l'ingresso di controllo Write vale 1
 - Usato quando il valore memorizzato è richiesto dopo



Metodologia di temporizzazione

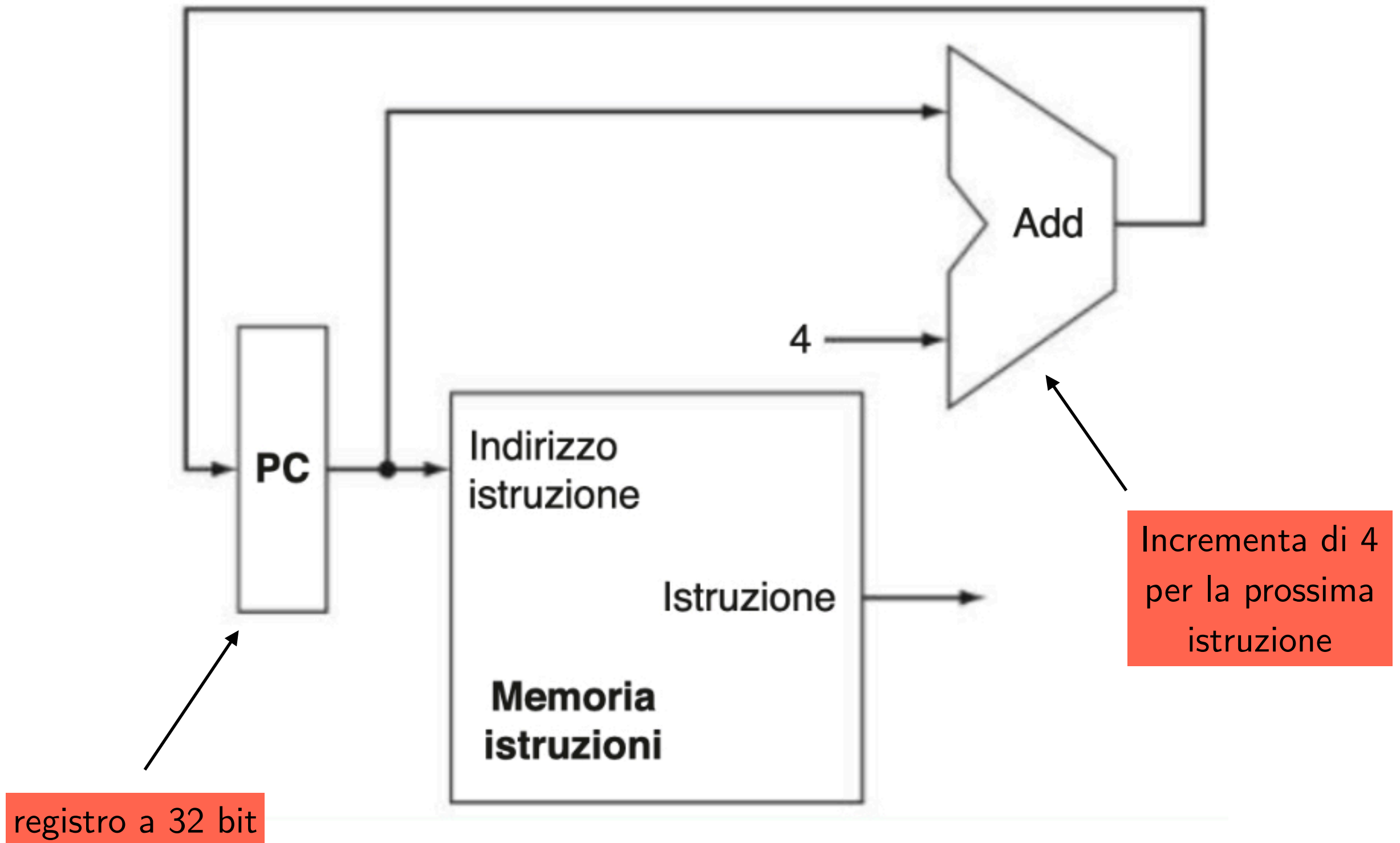
- La logica combinatoria trasforma i dati durante i cicli di clock
- Tra due fronti del clock
- Ingresso da elementi di stato, uscita a elementi di stato
- L'operazione più lunga determina il periodo di clock



Realizzare una unità di elaborazione

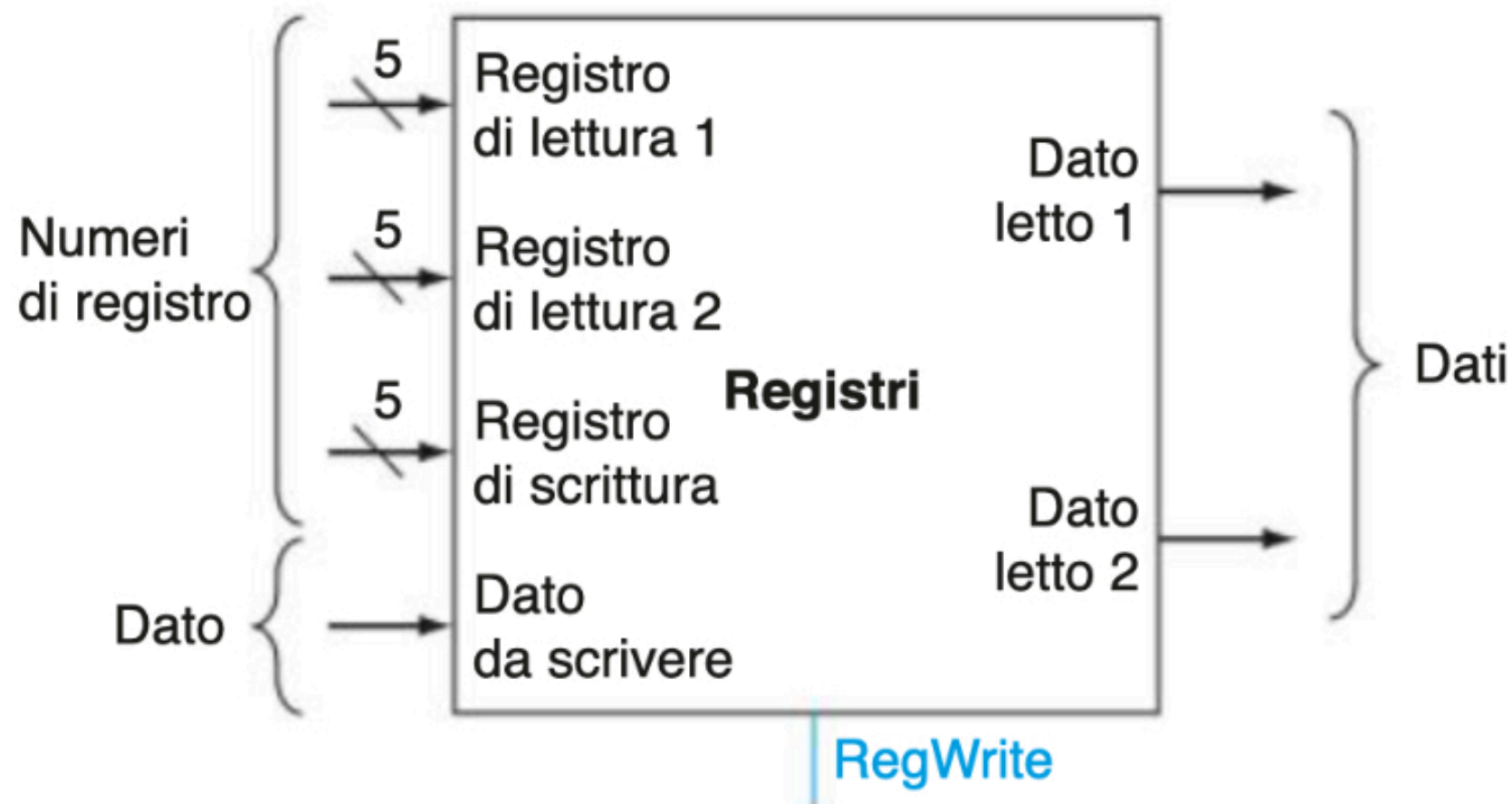
- Unità di elaborazione (datapath)
 - Elementi che elaborano i dati e gli indirizzi nella CPU
 - Registri, ALU, multiplexer, memorie, ...
- Costruiremo l'unità di elaborazione del MIPS incrementalmente
 - Rielaborando la panoramica di progetto

Fetch di un'istruzione

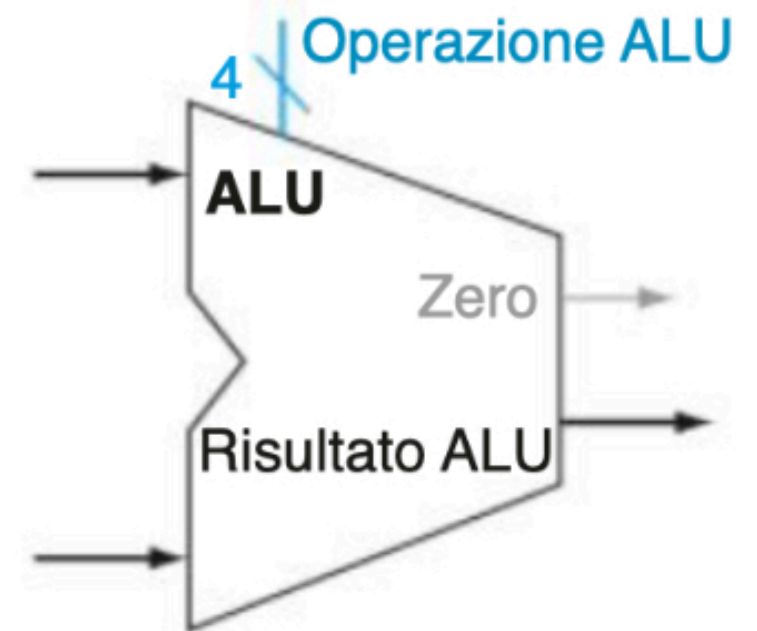


Istruzioni in formato R

- Leggere due operandi da registri
- Eseguire l'operazione aritmetico/logica
- Scrivere il risultato in un registro



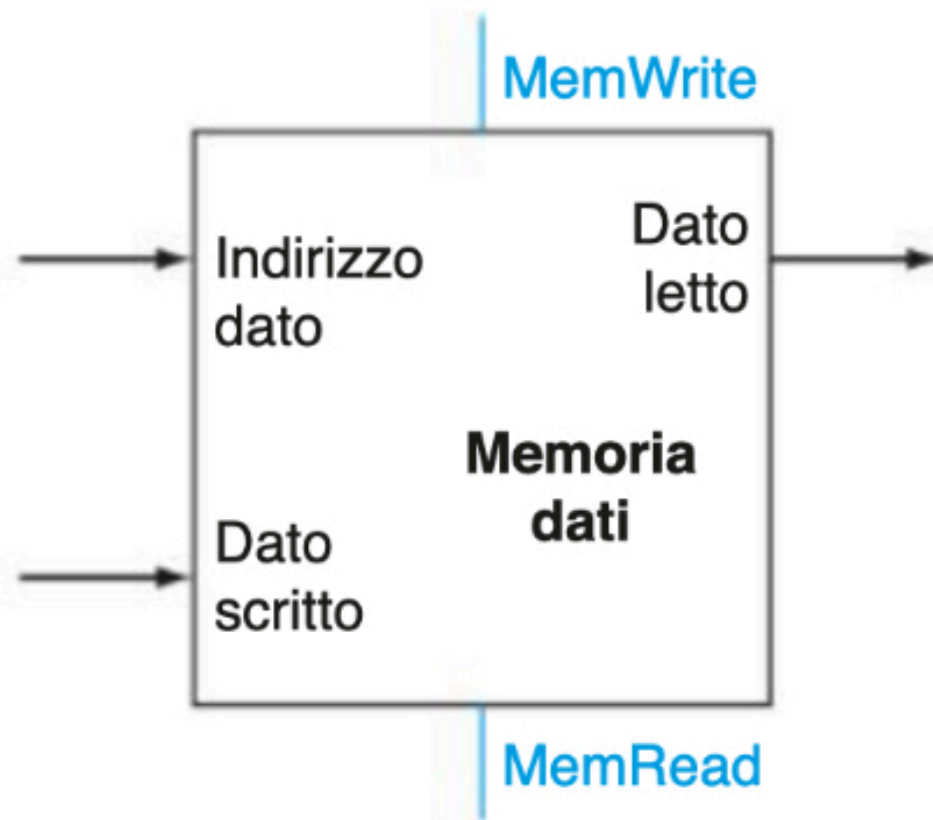
a. Registri



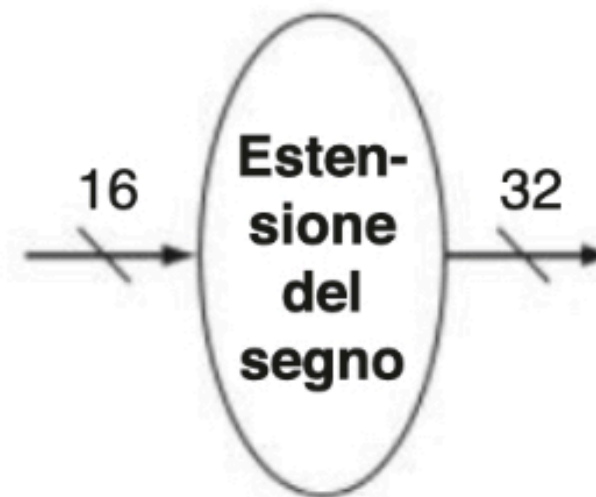
b. ALU

Istruzioni load/store

- Leggere operandi dei registri
- Calcolare l'indirizzo usando l'offset su 16 bit
 - Usando l'ALU, ma con offset esteso in segno
- Load: leggere dalla memoria e aggiornare registro
- Store: leggere un valore dal registro e copiarlo in memoria



a. Unità di memoria dati

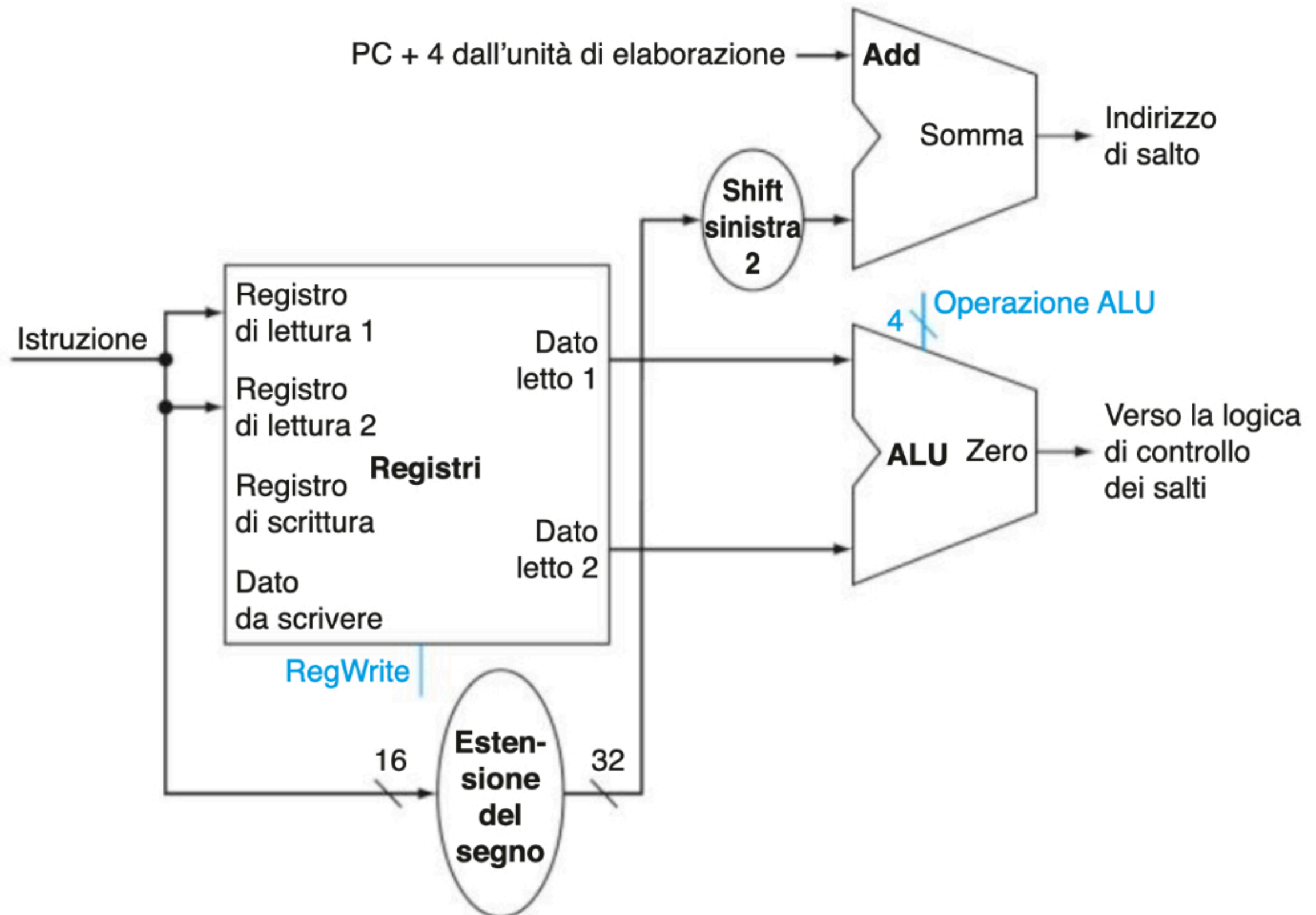


b. Unità di estensione del segno

Istruzioni di salto

- Leggere operandi dei registri
- Confrontare gli operandi
 - Usare l'ALU, sottrarre e controllare l'output Zero
- Calcolare l'indirizzo di destinazione
 - Spiazzamento esteso in segno
 - Scorrimento a sinistra di 2 bit (spiazzamento di parola)
 - Sommare a $PC + 4$
 - Già calcolato dal fetch dell'istruzione

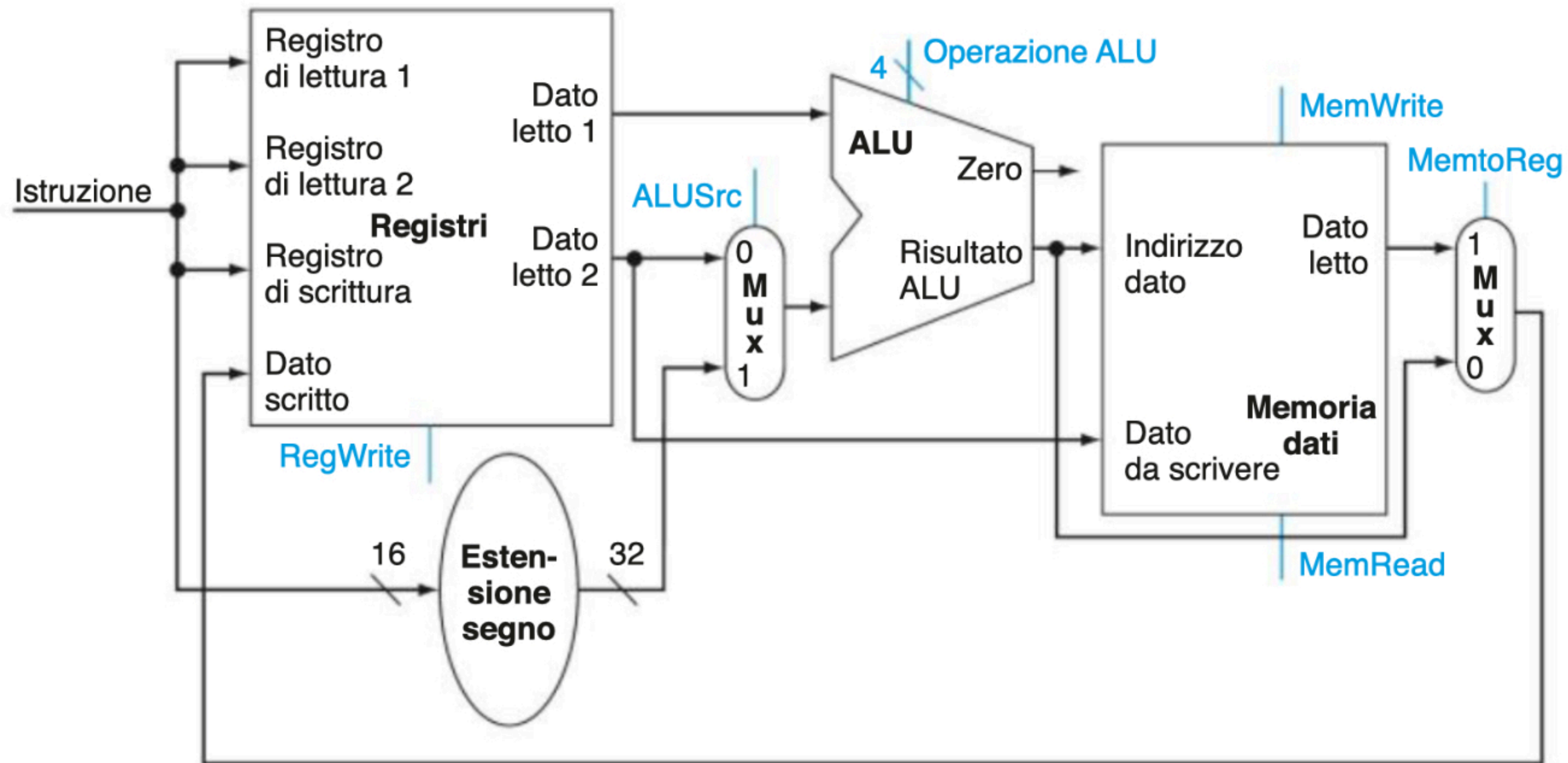
Istruzioni di salto



Comporre gli elementi

- L'unità di elaborazione più semplice cercherà di eseguire tutte le istruzioni in un singolo ciclo di clock
 - Nessuna risorsa dell'unità di elaborazione potrà essere utilizzata più di una volta per ogni istruzione
 - Quindi servirà quindi una memoria delle istruzioni separata da quella dei dati
- Usare multiplexer quando sorgenti alternative di dati sono usate per istruzioni differenti

Datapath formato R/load/store



Datapath completo

